

**KONRAD
LORENZ**
FUNDACIÓN UNIVERSITARIA





Electiva I: Backend para Web, Móvil e IoT

Tema 3: Ruby and Rails

Profesor: Jorge Iván Andrés Contreras Pereira

Agenda



Agenda

- Características de Rails
- Modelo Vista Controlador
- Archivos ERB
- CRUDL
- Relaciones
- Migraciones

¿Qué es Ruby and Rails?



Ruby - Rails

- Lenguaje de programación interpretado
- Todo son objetos
- Tipado dinámico
- Sintaxis limpia e intuitiva
- RubyGems: gestor de paquetes (gemas)
- “to enjoy programming, and to be happy”
- Framework MVC para Ruby
- Incluye ORM (mapeador modelos - tablas)
- Open Source. Creado en 2004
- “Convention over configuration”
- “Don’t repeat yourself”

Características



No hay corchetes {} en métodos o ifs

```
def hacer_positivo(numero)
  if numero < 0
    -numero
  else
    numero
  end
end
```

Return implícito

If y unless al final de línea

```
puts 'es par' if number.even?  
puts 'es par' unless number.odd?
```

Se pueden hacer llamadas a métodos sin paréntesis

```
resultado = misterio param1, param2  
resultado = misterio(param1, param2)  
resultado = misterio(param1, param2).otro_metodo
```

- `:simbolo`
- No son variables, No son constantes
- Podemos verlos como tags o etiquetas
- Fundamentalmente los usamos como índices en arrays asociativos (Hashes)
- A pesar de todo: También son objetos

```
def misterio(parametro)
  puts "Color: " + parametro[:color]
  puts "Ciudad: " + parametro[:ciudad]
end

misterio :color=>"rojo",:ciudad => "San Gil"
```

Bloques de código

Podemos pasar parámetros a un método

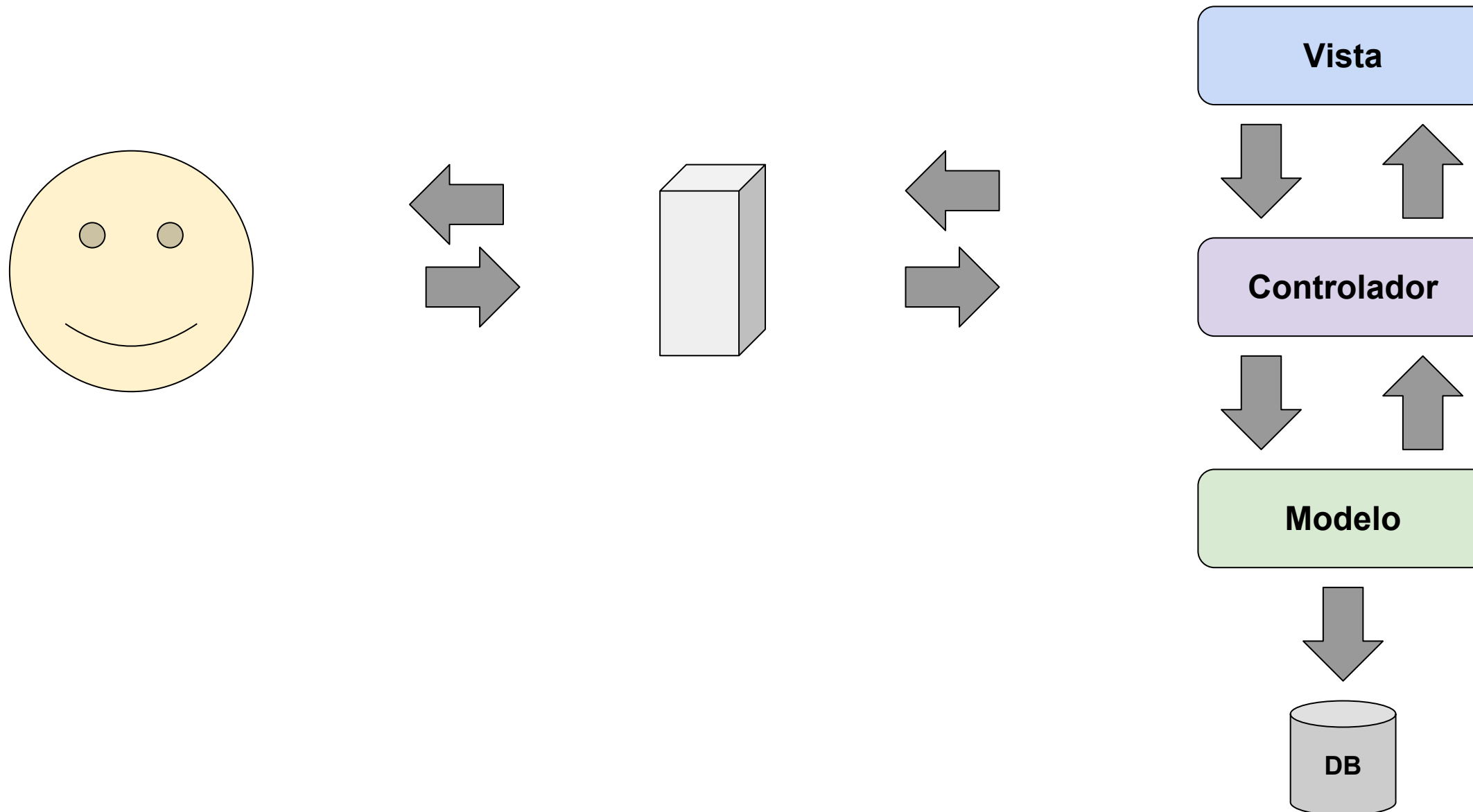
```
numeros = [1,2,3,4,5]

numeros.each {|numero| puts numero}
# Es equivalente a
numeros.each do |numero|
|  puts numero|
end
```

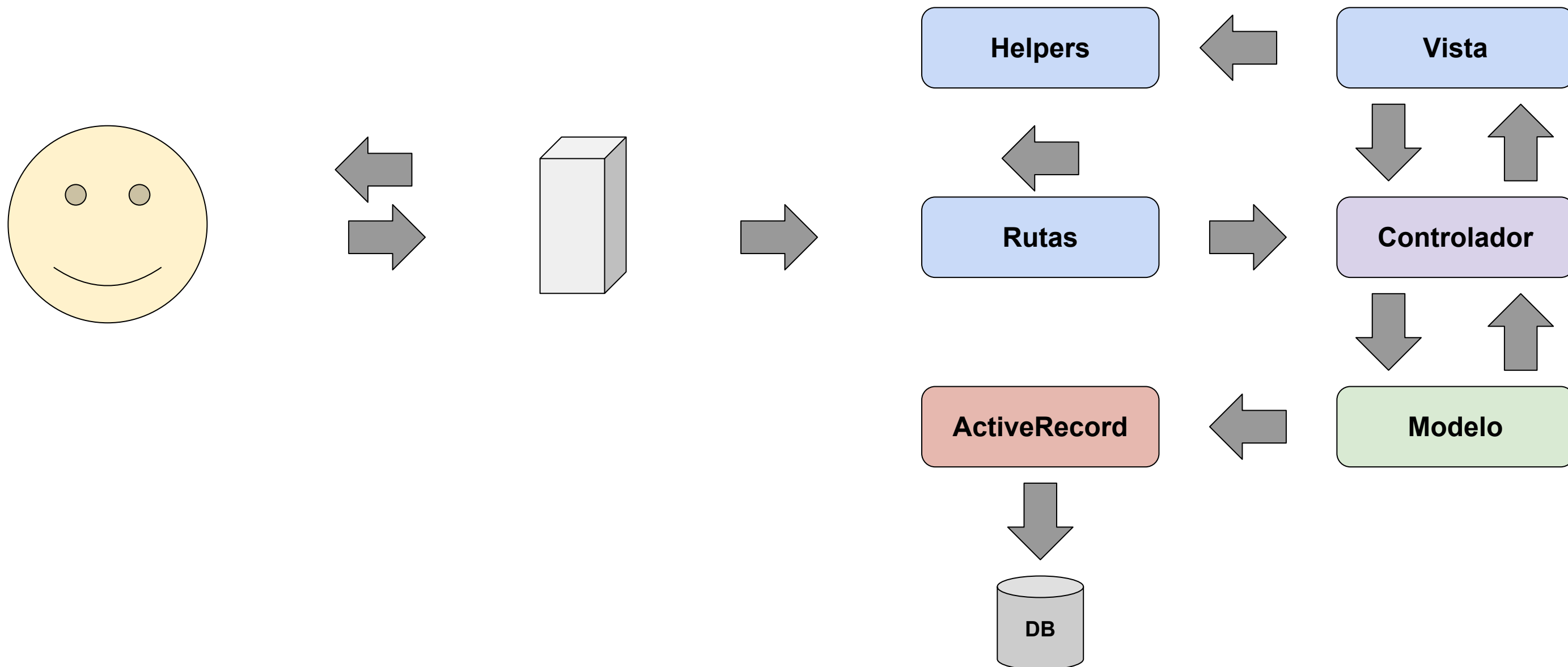
Modelo Vista Controlador



MVC



MVC en Rails



```
class User < ApplicationRecord
  include Clearance::User

  #Relaciones
  belongs_to :city
  belongs_to :institution

  has_many :companies
  has_many :activities

  has_one :student

  #Validaciones
  validates :role, presence: { message: "Para crear una cuenta debe asignarse un rol" }

  enum role: { superadmin: 0, admin: 1, digitador: 2, medico: 3, analista: 4, estudiante: 5,

  #Metodos de la clase
  def self.create_profile
    #...
  end

end
```



```
def index
  @users = User.all

  respond_to do |format|
    format.html
    format.js { render json: @user }
  end
end
```

Archivos ERB



Archivos ERB

```
<h1>Academic Informations</h1>

<table>
  <thead>
    <tr>
      <th>Student</th>
      <th>Title</th>
      <th>Year</th>
      <th>Institute</th>
      <th>City</th>
      <th>Country</th>
      <th colspan="3"></th>
    </tr>
  </thead>

  <tbody>
    <%= # Recorremos todos los elementos %>
    <% @academic_informations.each do |academic_information| %>
      <tr>
        <td><%= academic_information.student %></td>
        <td><%= academic_information.title %></td>
        <td><%= academic_information.year %></td>
        <td><%= academic_information.institute %></td>
        <td><%= academic_information.city %></td>
        <td><%= academic_information.country %></td>
        <td><%= link_to 'Show', academic_information %></td>
        <td><%= link_to 'Edit', edit_academic_information_path(academic_information) %></td>
        <td><%= link_to 'Destroy', academic_information, method: :delete, data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>

<br>

<%= link_to 'New Academic Information', new_academic_information_path %>
```

<% Código Ruby%>

<%= Expresión Ruby%>

<%# Comentario Ruby%>

Manejo de CRUDL



Las 5 operaciones básicas que se realizan sobre un objeto

- **C**reate (insertar objeto)
- **R**ead (leer objeto)
- **U**ppdate (actualizar objeto)
- **D**elete (borrar objeto)
- **L**ist (lista objetos)

Generación de una api

Usamos un modelo API/Rest

HTTP Verb	Ruta	Action	Usado para
GET	/users	index	Mostrar la lista de todos los usuarios
GET	/users/new	new	Una vista de un formulario de creación de usuarios
POST	/users	create	Crear un usuario
GET	/users/:id	show	Mostrar un usuario puntual
GET	/users/:id/edit	edit	Una vista de un formulario de edición de usuarios
PUT	/users/:id	update	Actualizar un usuario
DELETE	/users/:id	destroy	Borrar un usuario

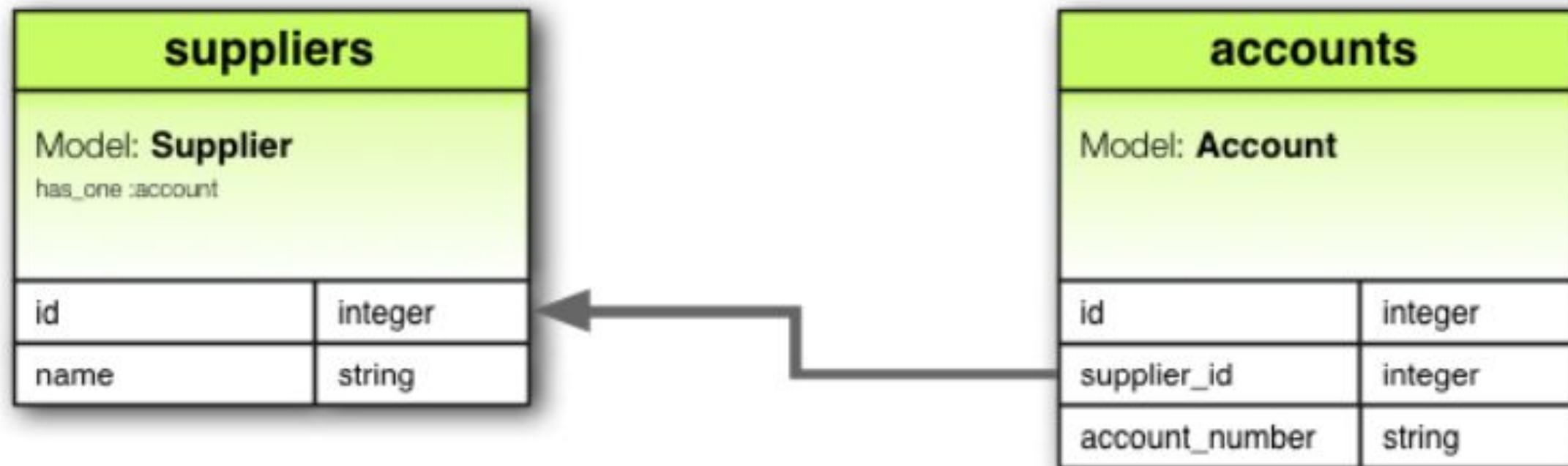
Relaciones



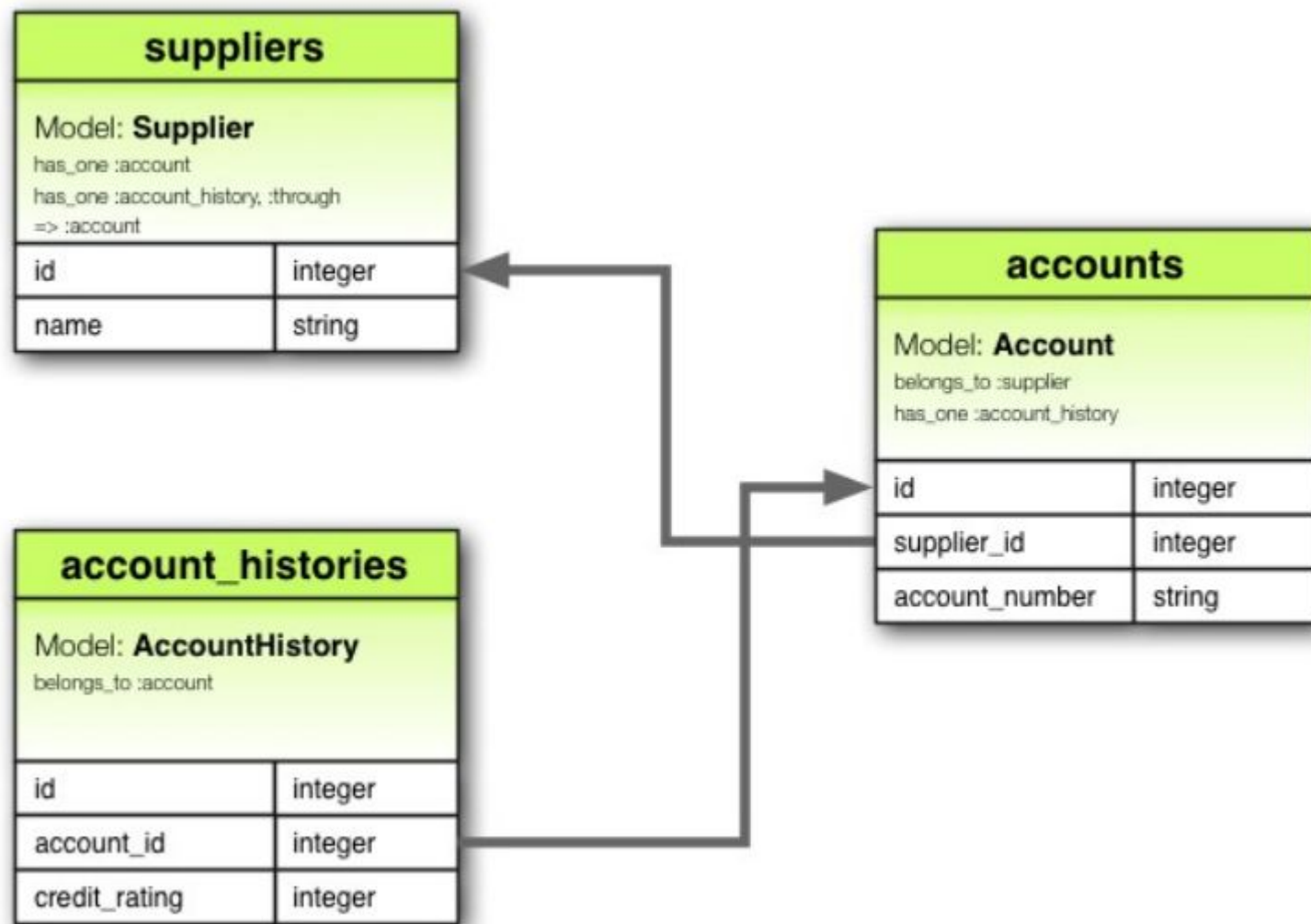
1 a 1



Relaciones 1a 1



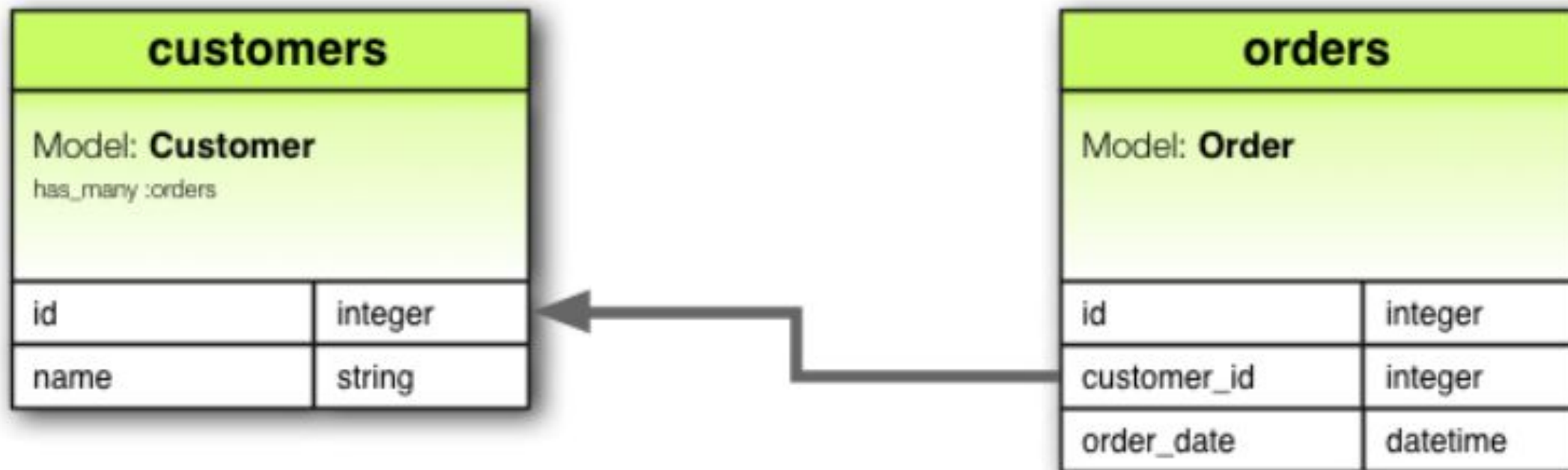
```
class Supplier < ActiveRecord::Base
  has_one :account
end
```



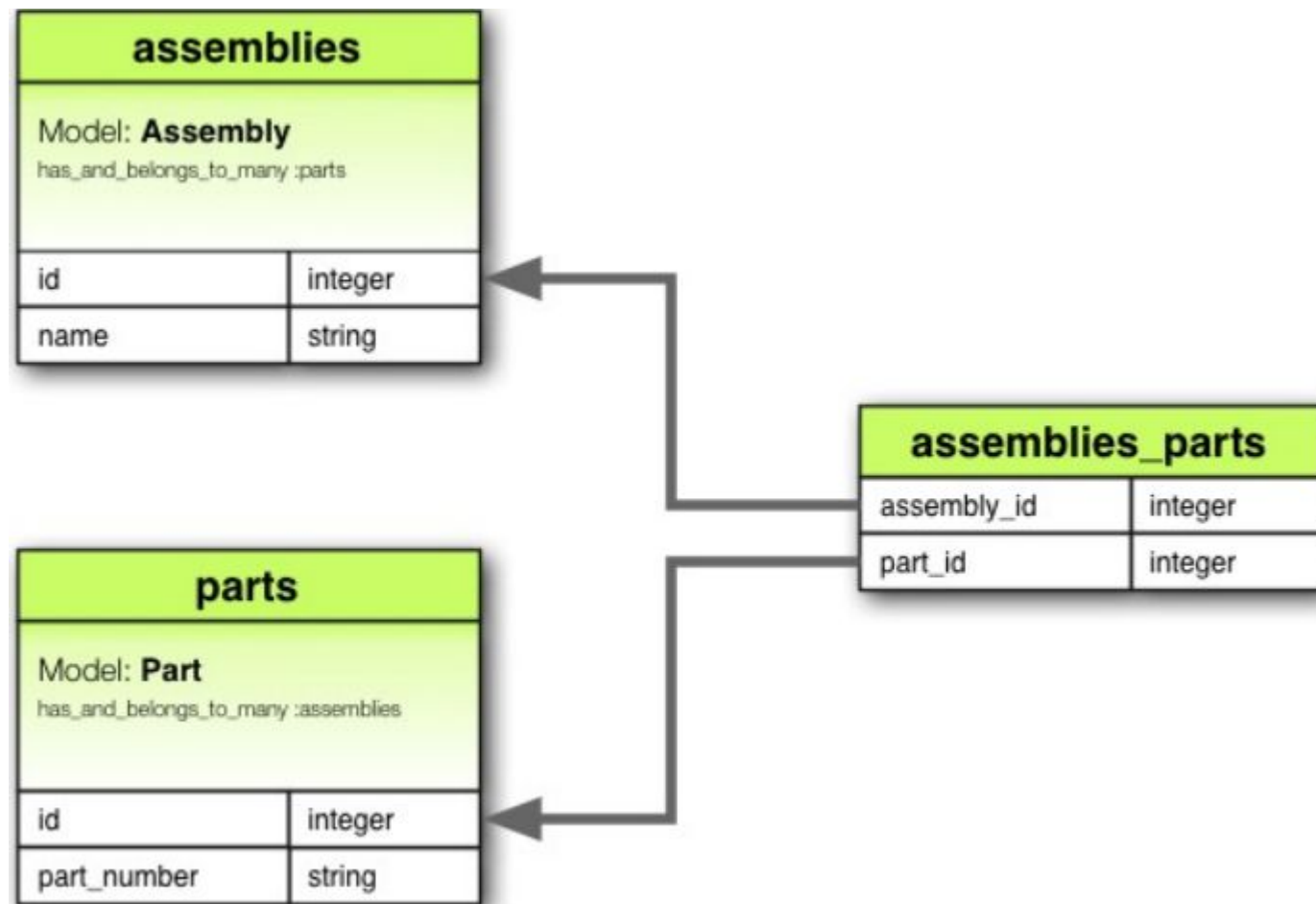
- 1 supplier tiene 1 account
- 1 supplier tiene 1 account_history a través de account
- 1 account pertenece a 1 supplier
- 1 account_history pertenece a 1 supplier

1 a N





```
class Customer < ActiveRecord::Base
  has_many :orders
end
```



Hay 3 tablas pero sólo dos modelos

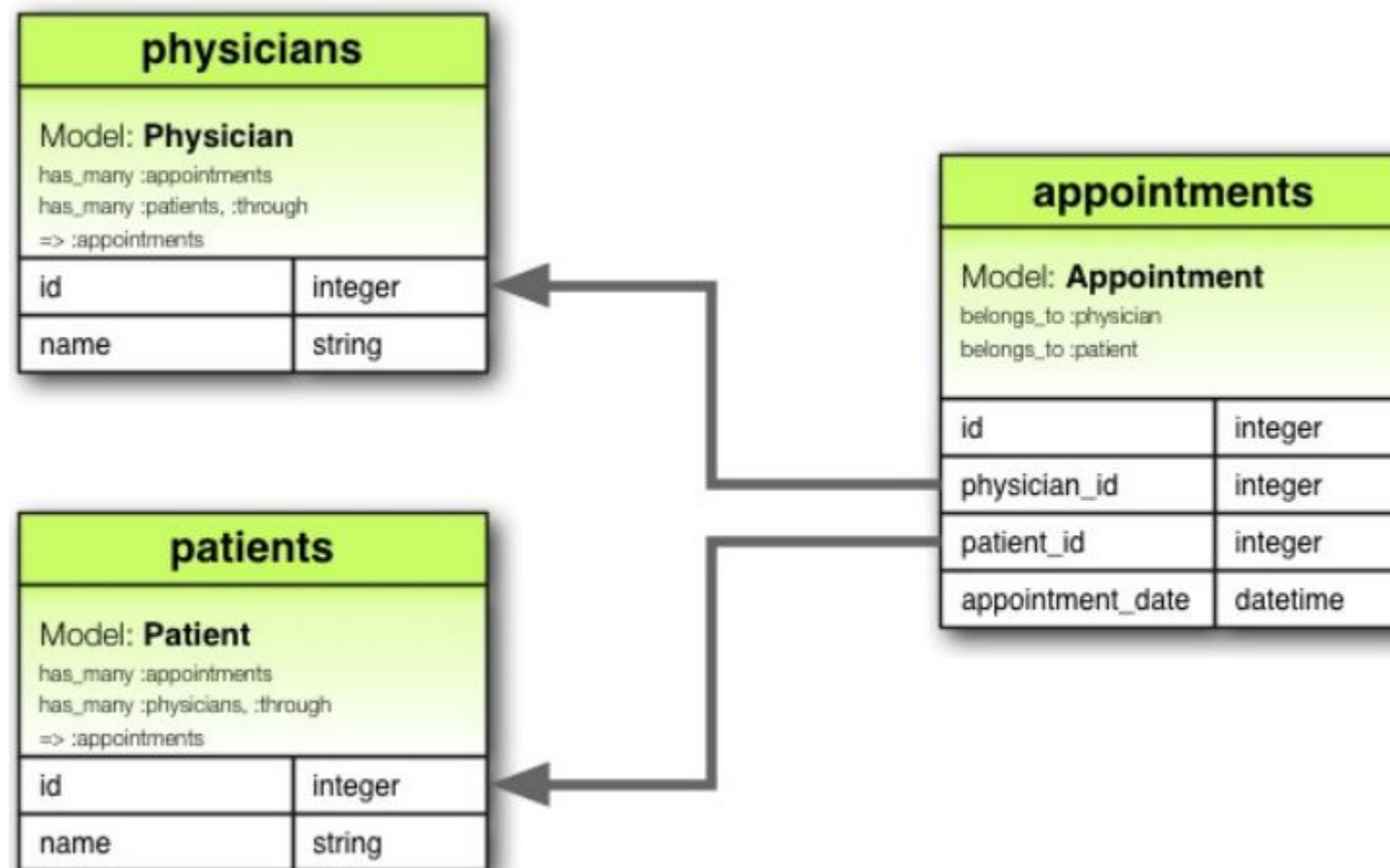
```
class Assembly < ActiveRecord::Base
  has_and_belongs_to_many :parts
end
```

```
class Part < ActiveRecord::Base
  has_and_belongs_to_many :assemblies
end
```

NaN



3 tablas con 3 modelos



```
class Physician < ActiveRecord::Base
  has_many :appointments
  has_many :patients, :through => :appointments
end
```

```
class Appointment < ActiveRecord::Base
  belongs_to :physician
  belongs_to :patient
end
```

```
class Patient < ActiveRecord::Base
  has_many :appointments
  has_many :physicians, :through => :appointments
end
```

Migraciones



- En Rails todas las modificaciones de la estructura de la DB deben hacerse a través de migraciones
- Las migraciones son archivos Ruby almacenados en [app]/db/migrate
- El nombre de archivo de cada migración lleva un timestamp que da orden de ejecución
- Permite crear o modificar tablas, columnas, índices u otros
- Facilitan la colaboración entre programadores

```
class CreateUsers < ActiveRecord::Migration[5.1]
  def change
    create_table :users do |t|
      t.timestamps null: false
      t.string :email, null: false, unique: true
      t.string :firstname, null: false
      t.string :othersname
      t.string :lastname, null: false
      t.string :otherslastname
      t.string :jobtitle
      t.integer :type_id
      t.bigint :citizen_number, unique: true
      t.string :encrypted_password, limit: 128, null: false
      t.string :confirmation_token, limit: 128
      t.string :remember_token, limit: 128, null: false
      t.references :institution, foreign_key: true
      t.references :city, foreign_key: true
      t.string :attachment
      t.boolean :term, null: false, default: false
    end

    add_index :users, :email
    add_index :users, :remember_token
  end
end
```

- El nombre de la tabla debe ir en plural
- El nombre del campo en singular

Crear migración

rails generate migration CreatePosts

Ejecutar migración

rake db:migrate

Deshacer cambios de la última migración

rake db:rollback

Ejemplo de Scaffolding



Crear una aplicación

rails new nombre

cd nombre

Arrancamos el servidor

rails server

Podremos consultar el proyecto en <http://localhost:3000>

Scaffolding tool



Un **Scaffold** nos permite generar automáticamente Modelo, Controlador, Migraciones, etc para un Modelo.

*rails generate scaffold Student firstname:string
lastname:string profession:integer
type_id:integer number_id:bigint birthdate:date
city:references nationality:string gender:integer
civil_status:integer profile:string
possibility_of_traveling:boolean interest:string
active:boolean*

```
jiacontrerasp@DESKTOP-5RL5338:/mnt/c/Users/jiacontrerasp/Development/jiacp/test/workshop$ rake db:migrate
== 20180803222741 CreateStudents: migrating =====
-- create_table(:students)
-> 0.0035s
== 20180803222741 CreateStudents: migrated (0.0047s) =====
```

```
Running via Spring preloader in process 10599
invoke active_record
create db/migrate/20180803222741_create_students.rb
create app/models/student.rb
invoke test_unit
create test/models/student_test.rb
create test/fixtures/students.yml
invoke resource_route
route resources :students
invoke scaffold_controller
create app/controllers/students_controller.rb
invoke erb
create app/views/students
create app/views/students/index.html.erb
create app/views/students/edit.html.erb
create app/views/students/show.html.erb
create app/views/students/new.html.erb
create app/views/students/_form.html.erb
invoke test_unit
create test/controllers/students_controller_test.rb
create test/system/students_test.rb
invoke helper
create app/helpers/students_helper.rb
invoke test_unit
invoke jbuilder
create app/views/students/index.json.jbuilder
create app/views/students/show.json.jbuilder
create app/views/students/_student.json.jbuilder
invoke assets
invoke coffee
create app/assets/javascripts/students.coffee
invoke scss
create app/assets/stylesheets/students.scss
invoke scss
create app/assets/stylesheets/scaffolds.scss
```

New Student

Firstname

Lastname

Profession

Type

Number

Birthdate
2018 ▼ August ▼ 3 ▼

Nationality

Gender

Civil status

Profile

Possibility of traveling
☐

Interest

Active
☐

Create Student

[Back](#)

Students

Firstname Lastname Profession Type Number Birthdate Nationality Gender Civil status Profile Possibility of traveling Interest Active

[New Student](#)

Consola



Entrar a la consola
cd nombre
rails console

Comandos

User.all

User.all.first

User.find(id)

User.find(id).inspect

User.find(id).relacion

User.where(id:1)



Gracias